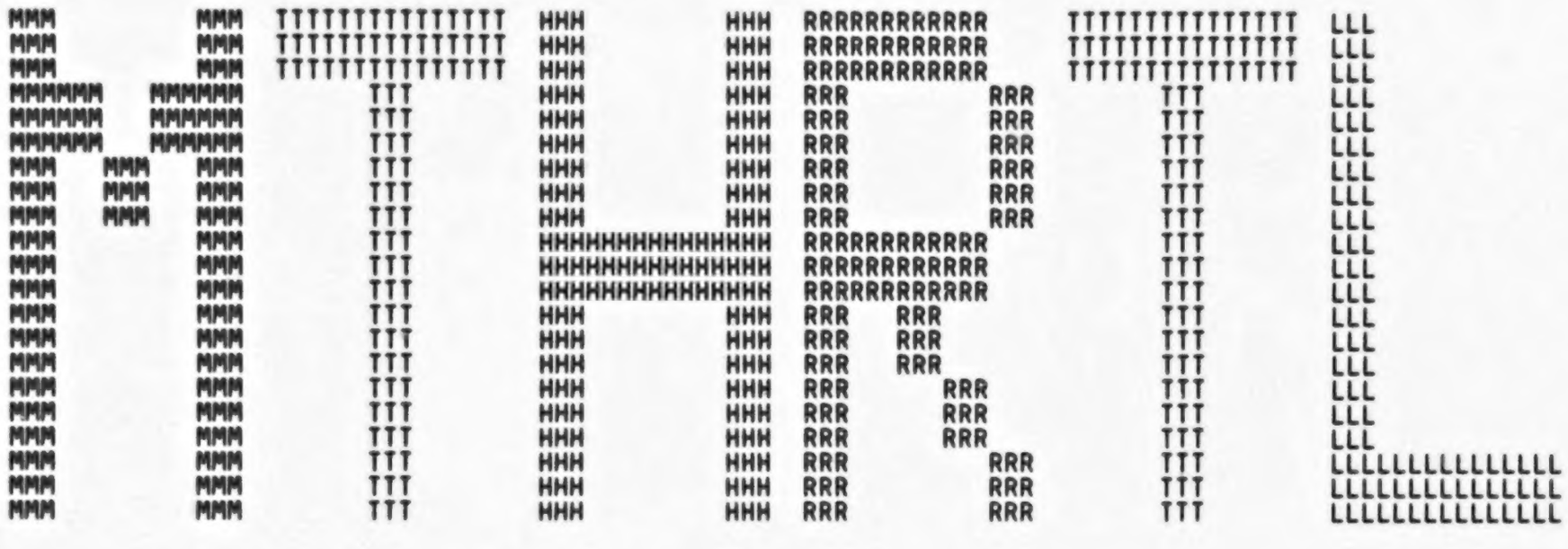
MMM MMM TTTTTTTTTTTTTTT HHH HHH RRRRRRRRRRR TTTTTTTTTTTTTTT LLL
MMMMMM MMMMMM TTT HHH HHH RRR RRR TTT LLL
MMM MMM MMM TTT HHH HHH RRR RRR TTT LLL
MMM MMM MMM TTT HHHHHHHHHHHHHHH RRRRRRRRRRR TTT LLL
MMM MMM MMM TTT HHH HHH RRR RRR TTT LLL
MMM MMM MMM TTT HHH HHH RRR RRR TTT LLL
MMM MMM MMM TTT HHH HHH RRR RRR TTT LLLLLLLLLLLLLLL

FILEID**OTSPOWCC

```
  OOOOOO    TTTTTTTTTT    SSSSSSSS  PPPPPPPP      OOOOOO    WW      WW    CCCCCCCC    CCCCCCCC
  OOOOOO    TTTTTTTTTT    SSSSSSSS  PPPPPPPP      OOOOOO    WW      WW    CCCCCCCC    CCCCCCCC
OO      OO      TT       SS        PP      PP  OO      OO  WW      WW  CC          CC
OO      OO      TT       SS        PP      PP  OO      OO  WW      WW  CC          CC
OO      OO      TT       SS        PP      PP  OO      OO  WW      WW  CC          CC
OO      OO      TT       SS        PP      PP  OO      OO  WW      WW  CC          CC
OO      OO      TT         SSSSSS  PPPPPPPP    OO      OO  WW      WW  CC          CC
OO      OO      TT         SSSSSS  PPPPPPPP    OO      OO  WW      WW  CC          CC
OO      OO      TT               SS PP          OO      OO  WW  WW  WW  CC          CC
OO      OO      TT               SS PP          OO      OO  WW  WW  WW  CC          CC
OO      OO      TT               SS PP          OO      OO  WWWW  WWWW  CC          CC                ....
OO      OO      TT               SS PP          OO      OO  WWWW  WWWW  CC          CC                ....
  OOOOOO        TT       SSSSSSSS  PP            OOOOOO    WW      WW    CCCCCCCC    CCCCCCCC          ....
  OOOOOO        TT       SSSSSSSS  PP            OOOOOO    WW      WW    CCCCCCCC    CCCCCCCC          ....

LL              IIIIII     SSSSSSSS
LL              IIIIII     SSSSSSSS
LL                II     SS
LL                II     SS
LL                II     SS
LL                II     SS
LL                II       SSSSSS
LL                II       SSSSSS
LL                II             SS
LL                II             SS
LL                II             SS
LL                II             SS
LLLLLLLLLL      IIIIII     SSSSSSSS
LLLLLLLLLL      IIIIII     SSSSSSSS
```

Table of contents

```
0000     1          .TITLE  OTS$POWCC - COMPLEX*8 ** COMPLEX*8 power routine
0000     2          .IDENT  /1-004/                 ; File: OTSPOWCC.MAR
0000     3
0000     4 ;
0000     5 ;*****************************************************************************
0000     6 ;*                                                                           *
0000     7 ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     8 ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     9 ;*  ALL RIGHTS RESERVED.                                                     *
0000    10 ;*                                                                           *
0000    11 ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
0000    12 ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    13 ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
0000    14 ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
0000    15 ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
0000    16 ;*  TRANSFERRED.                                                             *
0000    17 ;*                                                                           *
0000    18 ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
0000    19 ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    20 ;*  CORPORATION.                                                             *
0000    21 ;*                                                                           *
0000    22 ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
0000    23 ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    24 ;*                                                                           *
0000    25 ;*                                                                           *
0000    26 ;*****************************************************************************
0000    27 ;
0000    28
0000    29 ;++
0000    30 ; FACILITY: Language support library - user callable
0000    31 ;
0000    32 ; ABSTRACT:
0000    33 ;
0000    34 ;       COMPLEX*8 base to COMPLEX*8 power giving COMPLEX*8 result.
0000    35 ;
0000    36 ; ENVIRONMENT: User Mode, AST Reentrant
0000    37 ;
0000    38 ;--
0000    39 ; AUTHOR: Steven B. Lionel, CREATION DATE: 24-Oct-1978: Version 0
0000    40 ;
0000    41 ; MODIFIED BY:
0000    42 ;
0000    43 ; SBL 24-Oct-1978, VERSION 00
0000    44 ; 1-001 - Original
0000    45 ; 1-002 - Standardized version number format, with three digits in
0000    46 ;            the edit number.  JBS 16-NOV-78
0000    47 ; 1-003 - Add "_" to the PSECT directive.  JBS 22-DEC-78
0000    48 ; 1-004 - Change shared external references to G^ RNH 25-Sep-81
```

```
                     0000     50          .SBTTL  DECLARATIONS
                     0000     51 ;
                     0000     52 ; INCLUDE FILES:
                     0000     53 ;
                     0000     54
                     0000     55 ;
                     0000     56 ; EXTERNAL DECLARATIONS:
                     0000     57 ;
                     0000     58          .GLOBL  MTH$CEXP                ; Complex exponentiation
                     0000     59          .GLOBL  MTH$CLOG                ; Complex logarithm
                     0000     60 ;
                     0000     61 ; MACROS:
                     0000     62 ;
                     0000     63
                     0000     64 ;
                     0000     65 ; EQUATED SYMBOLS:
                     0000     66 ;
00000004             0000     67          base = 4                        ; base input - by value
0000000C             0000     68          exp = 12                        ; exponent input - by value
                     0000     69 ;
                     0000     70 ; OWN STORAGE:
                     0000     71 ;
                     0000     72
                     0000     73 ;
                     0000     74 ; PSECT DECLARATIONS:
                     0000     75 ;
                 00000000     76          .PSECT _OTS$CODE PIC, USR, CON, REL, LCL, SHR, -
                     0000     77                      EXE, RD, NOWRT, LONG
                     0000     78
```

```
                        .SBTTL  OTS$POWCC  - COMPLEX*8 ** COMPLEX*8 power routine
                ;++
                ; FUNCTIONAL DESCRIPTION:
                ;
                ;       OTS$POWCC evaluates the result of taking a complex base
                ;       to a complex power.  The ANS FORTRAN X3.9-1978 standard defines
                ;       complex exponentiation as:
                ;
                ;       x ** y = CEXP( y * CLOG(x))
                ;
                ;       where x and y are type COMPLEX.
                ;
                ;       The arguments of OTS$POWCC are CALL BY VALUE.
                ;
                ; CALLING SEQUENCE:
                ;
                ;       power.wfc.v = OTS$POWCC (base.rfc.v, exponent.rfc.v)
                ;
                ; INPUT PARAMETERS:
                ;
                ;       Both base and exponent are COMPLEX*8 numbers, each consisting
                ;       of a REAL*4 real part and a REAL*4 imaginary part.  Both are
                ;       CALL BY VALUE.
                ;
                ; IMPLICIT INPUTS:
                ;
                ;       NONE
                ;
                ; OUTPUT PARAMETERS:
                ;
                ;       NONE
                ;
                ; IMPLICIT OUTPUTS:
                ;
                ;       NONE
                ;
                ; FUNCTION VALUE:
                ;
                ;       The COMPLEX*8 (REAL*4, REAL*4) result of taking the
                ;       COMPLEX base to the COMPLEX exponent power.
                ;
                ; SIDE EFFECTS:
                ;
                ;       Possible error signals are:
                ;
                ;        MTH$_INVARGMAT if base is (0.,0.).
                ;        MTH$_FLOOVEMAT if floating overflow occurs.
                ;        MTH$_SINCOSSIG if absolute value of the imaginary part of
                ;                       (exponent * CLOG(base)) > PI*2**30.
                ;        SS$_ROPRAND    if reserved floating operand is fetched.
                ;--
```

```
                     003C  0000   132          .ENTRY  OTS$POWCC, ^M<R2,R3,R4,R5>      ; disable integer ovflo
                           0002   133
                           0002   134          MTH$FLAG_JACKET                         ; establish math error handler
                           0002
     6D  00000000'GF   9E  0002                MOVAB   G^MTH$$JACKET_HND, (FP)
                           0009                                                        ; set handler address to jacket
                           0009                                                        ; handler
                           0009
                           0009   135
          52   0C AC   7D  0009   136          MOVQ    exp(AP), R2                     ; put exponent in R2, R3
                           000D   137                                                  ; later operations will check
                           000D   138                                                  ; for reserved operands
                           000D   139
                           000D   140 ;+
                           000D   141 ;       Get complex logarithm of base
                           000D   142 ;-
                04 AC   DF  000D   143          PUSHAL  base(AP)                        ; address of base
     00000000'GF    01   FB  0010   144          CALLS   #1, G^MTH$CLOG                  ; R0,R1 get LOG(base)
                           0017   145                                                  ; call by reference
                           0017   146
                           0017   147 ;+
                           0017   148 ;       CLOG(base) is in R0, R1.  Multiply by exponent.
                           0017   149 ;       R0,R1 = CLOG(base) = a+bi
                           0017   150 ;       R2,R3 = exp  = c+di
                           0017   151 ;
                           0017   152 ;       Complex multiplication defined as:
                           0017   153 ;
                           0017   154 ;       real part = ac-bd
                           0017   155 ;       imaginary part = ad+bc
                           0017   156 ;-
                           0017   157
     54   53   50   45   0017   158          MULF3   R0, R3, R4                      ; R4 = ad
          50   52   44   001B   159          MULF2   R2, R0                          ; R0 = ac
     55   53   51   45   001E   160          MULF3   R1, R3, R5                      ; R5 = bd
          50   55   42   0022   161          SUBF2   R5, R0                          ; R0 = ac-bd
          51   52   44   0025   162          MULF2   R2, R1                          ; R1 = bc
          51   54   40   0028   163          ADDF2   R4, R1                          ; R1 = ad+bc
                           002B   164
                           002B   165 ;+
                           002B   166 ;       Now compute CEXP(product)
                           002B   167 ;-
                           002B   168
          7E   50   7D   002B   169          MOVQ    R0, -(SP)                       ; put product (R0,R1) on stack
               5E   DD   002E   170          PUSHL   SP                              ; address of arguments
     00000000'GF    01   FB  0030   171          CALLS   #1, G^MTH$CEXP                  ; R0, R1 get EXP(product)
                           0037   172                                                  ; call by reference
                           0037   173
                     04    0037   174          RET                                     ; all done, exit
                           0038   175          .END
```

```
BASE            = 00000004
EXP             = 0000000C
MTH$$JACKET_HND   ********   X   01
MTH$CEXP          ********   G   00
MTH$CLOG          ********   G   00
OTS$POWCC         00000000  RG   01
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
_OTS$CODE	00000038 (56.)	01 (1.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	32	00:00:00.08	00:00:01.80
Command processing	138	00:00:00.59	00:00:05.14
Pass 1	87	00:00:00.65	00:00:03.71
Symbol table sort	0	00:00:00.00	00:00:00.00
Pass 2	45	00:00:00.48	00:00:04.60
Symbol table output	2	00:00:00.01	00:00:00.05
Psect synopsis output	2	00:00:00.02	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	308	00:00:01.85	00:00:15.34

```
The working set limit was 900 pages.
2597 bytes (6 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 6 non-local and 0 local symbols.
235 source lines were read in Pass 1, producing 11 object records in Pass 2.
1 page of virtual memory was used to define 1 macro.
```

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SYSLIB]STARLET.MLB;2	0

```
0 GETS were required to define 0 macros.

There were no errors, warnings or information messages.

MACRO/ENABLE=SUPPRESSION/DISABLE=(GLOBAL,TRACEBACK)/LIS=LIS$:OTSPOWCC/OBJ=OBJ$:OTSPOWCC MSRC$:MTHJACKET/UPDATE=(ENH$:MTHJACKET)+MSRC
```

OTSMULCD
LIS

OTSPOWCGC
LIS

OTSDIVC
LIS

OTSPOWDD
LIS

OTSPOWCC
LIS

OTSPOWGG
LIS

OTSPOWGJ
LIS

OTSPOWCDJ
LIS

MTHTAN
LIS

MTHVECTOR
LIS

OTSDIVCG
LIS

OTSPOWCJ
LIS

OTSPOWDLU
LIS

MTHTANH
LIS

OTSMULCG
LIS

OTSPOWCGJ
LIS

OTSPOWDJ
LIS

OTSDIVCD
LIS

OTSPOWCDC
LIS